Robust Real-time Intrusion Detection System

Byung-Joo Kim*, and Il-Kon Kim**

Abstract: Computer security has become a critical issue with the rapid development of business and other transaction systems over the Internet. The application of artificial intelligence, machine learning and data mining techniques to intrusion detection systems has been increasing recently. But most research is focused on improving the classification performance of a classifier. Selecting important features from input data leads to simplification of the problem, and faster and more accurate detection rates. Thus selecting important features is an important issue in intrusion detection. Another issue in intrusion detection is that most of the intrusion detection systems are performed by off-line and it is not a suitable method for a real-time intrusion detection system. In this paper, we develop the real-time intrusion detection system, which combines an on-line feature extraction method with the Least Squares Support Vector Machine classifier. Applying the proposed system to KDD CUP 99 data, experimental results show that it has a remarkable feature extraction and classification performance compared to existing off-line intrusion detection systems.

Keywords: real-time IDS, kernel PCA. LS-SVM

1. Introduction

Computer security has become a critical issue with the rapid development of business and other transaction systems over the Internet. Intrusion detection aims to detect intrusive activities while they are acting on computer network systems. Most intrusion detection systems (IDSs) are based on hand-crafted signatures that are developed by manual coding of expert knowledge. These systems match activity on the system being monitored to known signatures of attack. The major problem with this approach is that these IDSs fail to generalize to detect new attacks or attacks without known signatures. Recently, there has been an increased interest in data mining based approaches to building detection models for IDSs. These models generalize from both known attacks and normal behavior in order to detect unknown attacks. They can also be generated via a quicker and more automated method than manually encoded models that require difficult analysis of audit data by domain experts. Several effective data mining techniques for detecting intrusions have been developed [1][2][3], many of which perform close to or better than systems engineered by domain experts. However, successful data mining techniques are themselves not enough to create effective IDSs. Despite the promise of better detection performance and generalization ability of data mining based IDSs, there are some difficulties in the implementation of the system. We can group these difficulties into three general categories: accuracy (i.e., detection performance), efficiency, and usability. In this paper, we discuss the accuracy problem in developing a real-time two-tier based IDS. Another issue with an IDS is that it should operate in real-time. In typical applications of data mining to intrusion detection, detection models are produced off-line because the learning algorithms must process tremendous amounts of archived audit data. These models can naturally be used for off-line intrusion detection. An effective IDS should work in real-time, as intrusions take place, to minimize security compromises. Elimination of the insignificant and/or useless inputs leads to a simplification of the problem, and faster and more accurate detection results. Feature selection therefore is an important issue in intrusion detection. In this paper we present an on-line feature extraction method from audit data which helps discriminate between attacks and normal data. These features can then be used by any of the classification algorithms. Principal Component Analysis (PCA) [4] is a powerful technique for extracting features from data sets. For reviews of the existing literature see [5][6][7]. Traditional PCA, however, has several problems. First PCA requires a batch computation step and it causes a serious problem when the data set is large i.e., the PCA computation becomes very expensive. The second problem is that, in order to update the subspace of eigenvectors with other data, we have to recompute the whole eigenspace. The final problem is that PCA only defines a linear projection of the data; the scope of its application is necessarily somewhat limited. It has been shown that most of the data in the real world are inherently non-symmetrical and therefore contain higher-order correlation information that could be useful [8]. PCA is incapable of representing

Manuscript received September 29, 2005; accepted November 8, 2005.
This study was supported by a grant of the Korea Health 21 R&D Project, Ministry of Health & Welfare, Republic of Korea(02-PJ1-PG6-HI03-0004)

* Department of Information & Communication Engineering at Youngsan University, Pusan, Korea (bjkim@ysu.ac.kr)
** Department of Computer Science at Kyungpook National University Daegu, Korea (ikkim@kyungpook.ac.kr)

such data. For such cases, nonlinear transforms are necessary. Recently the kernel trick has been applied to PCA and is based on a formulation of PCA in terms of the dot product matrix instead of the covariance matrix [9]. Kernel PCA (KPCA), however, requires storing and finding the eigenvectors of an N X N kernel matrix where N is a number of patterns. It is an infeasible method when N is large. This fact has motivated the development of on-line KPCA method which does not store the kernel matrix.

It is hoped that the distribution of the extracted features in the feature space has a simple distribution so that a classifier can do a proper task. But it is pointed out that features extracted by KPCA are global features for all input data and thus may not be optimal for discriminating one class from others [9]. In order to solve this problem, we developed the two-tier intrusion detection system. Proposed real time IDS is composed of two parts. The first part is used for on-line feature extraction. To extract on-line nonlinear features, we propose a new feature extraction method which overcomes the problem of memory requirement of KPCA by an on-line eigenspace update method incorporating an adaptation of the kernel function. The second part is used for classification. Extracted features are used as input for classification. We take Least Squares Support Vector Machines (LS-SVM) [10] as a classifier. LS-SVM is reformulations to the standard Support Vector Machines (SVM) [11]. SVM typically solves problems by quadratic programming (QP). Solving the QP problem requires complicated computational effort and has more memory requirement. LS-SVM overcomes this problem by solving a set of linear equations in the problem formulation. This paper is composed as follows. In Section 2 we will briefly explain the incremental feature extraction method. In Section 3 KPCA is introduced and to make KPCA incrementally, the empirical kernel map method is explained. The proposed classifier combining LS-SVM with the proposed feature extraction method is described in Section 4. Experimental results to evaluate the performance of the proposed system are shown in Section 5. Discussion of the proposed IDS and future work are described in Section 6.

2. Incremental PCA

In this section we briefly outline the method that allows for complete incremental learning using the eigenspace approach. The method uses the incremental PCA algorithm and projects all input data immediately onto the subspace. Each input datum is then discarded, and its representation consists only of the corresponding principal components stored.

Let there be a data set with examples of dimension. We compute the eigensystem by solving the singular value decomposition (SVD) of the covariance matrix composed as

$$C = \frac{1}{N}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^T \tag{1}$$

where $\bar{x} = \frac{1}{N}\sum_{i=1}^{n} x_i$ is the mean input vector.

The eigenvectors $u_i, i = 1 \cdots N$ corresponding to non-zero eigenvectors of the covariance matrix span a subspace of maximum dimensions. We can then choose a subset of only eigenvectors corresponding to the largest eigenvalues to be included in the model. To explain incremental PCA, we assume we have already built a set of eigenvectors $U = [u_j], j = 1 \cdots k$, after having used the input data $x_i, i = 1 \cdots N$. The corresponding eigenvalues are $\lambda = diag(\Lambda)$ and $\bar{x}$ is the mean input vector. Incremental building requires updating of these eigenspaces to take into account the new input data x_{N+1}.

Here we briefly summarize the method described in [4]. First, we update the mean

$$\bar{x}' = \frac{1}{N+1}(N\bar{x} + x_{N+1}) \tag{2}$$

We then update the set of eigenvectors by adding a new vector and applying a rotational transformation. In order to do this, we first compute the orthogonal residual vector $\hat{h} = (Ua_{N+1} + \bar{x}) - x_{N+1}$ and normalize it to obtain $h_{N+1} = \frac{\hat{h}_{N+1}}{|\hat{h}_{N+1}|_2}$ for $|\hat{h}_{N+1}|_2 > 0$ and $h_{N+1} = 0$ otherwise. The new matrix of eigenvectors U' is computed by

$$U' = [U, h_{N+1}]R \tag{3}$$

where $R \in \mathbb{R}^{(k+1)\times(k+1)}$ is a rotation matrix. R is the solution of the eigenproblem of the following form

$$DR = R\Lambda' \tag{4}$$

We compose $D \in \mathbb{R}^{(k+1)\times(k+1)}$ as

$$D = \frac{N}{N+1}\begin{bmatrix}\Lambda & 0\\ 0^T & 0\end{bmatrix} + \frac{N}{(N+1)^2}\begin{bmatrix}aa^T & \gamma a\\ \gamma a^T & \gamma^2\end{bmatrix} \tag{5}$$

where $\gamma = h_{N+1}^T(x_{N+1} - \bar{x})$ and $a = U^T(x_{N+1} - \bar{x})$. There are other ways to construct the matrix D [3,14]. However, only the method described in [4] allows for the updating of the mean.

3. Incremental Kernel Principal Component Analysis (IKPCA)

Although incremental PCA builds the subspace of eigenvectors incrementally, it is limited to linear data. For the case of a nonlinear data set, applying the feature mapping function method to incremental PCA may be the solution. This is performed via the so-called *kernel-trick,* which is an implicit mapping to an infinite dimensional space.

$$K(x,y) = \Phi(x) \cdot \Phi(y) \tag{6}$$

Where K is a given the kernel function in input space. When K is semi positive definite, the existence of Φ is proven [15]. But, in most cases the mapping Φ cannot be obtained explicitly, so the vector in the feature space is not observable and only the inner product between vectors can be observed via the kernel function. However, for a given data set, it is possible to approximate Φ by the empirical kernel map proposed by Scholkop [15] and Tsuda [16], which is defined as $\Psi_N : \mathbb{R}^d \rightarrow \mathbb{R}^N$

$$\begin{aligned}\Psi_N(x) &= [\Phi(x_1) \cdot \Phi(x), \cdots, \Phi(x_N) \cdot \Phi(x)]^T \\ &= [k(x_1, x), \cdots, k(x_N, x)]^T\end{aligned}$$

A performance evaluation of the empirical kernel map was shown by Tsuda. He shows that a support vector machine with an empirical kernel map is identical to the conventional kernel map [17].

4. Proposed System

In the previous Section 3 we proposed an incremental KPCA method for nonlinear feature extraction. Feature extraction by incremental KPCA effectively acts as nonlinear mapping from the input space to an implicit high dimensional feature space. It is hoped that the distribution of the mapped data in the feature space have simple distribution so that a classifier can classify them properly. But it is pointed out that features extracted by KPCA are global features for all input data and thus may not be optimal for discriminating one class from others. For classification purposes, after global features are extracted using they must be used as input data for classification. There are many famous classifiers in the machine learning field. Among them the neural network is a popular method for classification and prediction purposes. Traditional neural network approaches, however have suffered difficulties with generalization, producing models that can overfit the data. To overcome the problem of classical neural network technique, support vector machines (SVM) have been introduced. The foundations of SVM have been developed by Vapnik and it is a powerful methodology for solving problems in nonlinear classification.

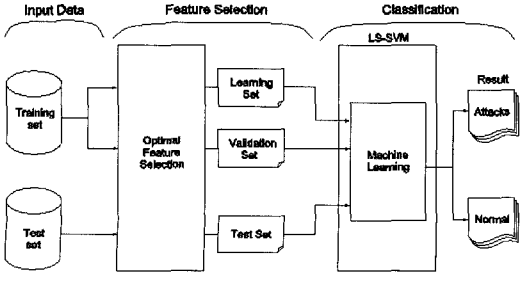

Fig. 1. Overall structure of real-time IDS

Originally, it was introduced within the context of statistical learning theory and structural risk minimization. In the methods one solves convex optimization problems, typically by quadratic programming (QP). Solving the QP problem requires complicated computational effort and more memory requirement. LS-SVM overcomes this problem by solving a set of linear equations in the problem formulation. The LS-SVM method is computationally attractive and easier to extend than SVM.

5. Experiment

To evaluate the performance of the proposed real-time IDS system, we use KDD CUP 99 data [17]. The following sections present the results of experiments.

5.1 Description of Dataset

The raw training data (kddcup.data.gz) was about four gigabytes of compressed binary TCP dump data from seven weeks of network traffic. This was processed into about five million connection records. Similarly, the two weeks of test data yielded around two million connection records. A connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows to and from a source IP address to a target IP address under some well defined protocol. Each connection is labeled as either normal, or as an attack, with exactly one specific attack type. Each connection record consists of about 100 bytes. Attacks fall into four main categories:

- DOS: denial-of-service, e.g. syn flood;
- R2L: unauthorized access from a remote machine, e.g. guessing password;
- U2R: unauthorized access to local superuser (root) privileges, e.g., various "buffer overflow" attacks;
- Probing: surveillance and other probing, e.g., port scanning.

It is important to note that the test data (corrected.gz) is not from the same probability distribution as the training data, and it includes specific attack types not in the training data. This makes the task more realistic. The datasets contain a total of 24 training attack types, with an additional 14 types in the test data only.

5.2 Experimental Condition

To evaluate the classification performance of the proposed system, we randomly split the training data as 80% and the remaining as validation data. To evaluate the classification accuracy of the proposed system we compare the proposed system to SVM. Because standard LS-SVM and SVM are only capable of binary classification, we take multi-class LS-SVM and SVM. An RBF kernel has been taken and the optimal hyper-parameter of multi-class SVM and LS-SVM [20] was obtained by 10-fold cross-validation

procedure. In [19] it is shown that the use of 10-fold cross-validation for hyper-parameter selection of SVM and LS-SVMs consistently leads to very good results. By experiment we will evaluate the generalization ability of the proposed IDS on the test data set since there are 14 additional attack types in the test data which are not included in the training set. To do this, extracted by on-line KPCA will be used as input for multi-class LS-SVM. Our results are summarized in the following sections.

5.3 Evaluate Feature Extraction Performance

Table 1 gives the results of extracted features for each class by incremental KPCA method.

Table 1. Extracted features of each class by incremental KPCA.

Class	Extracted features
Normal	1,2,3,5,6,7,8,9,10,11,12,13,14,16,17,18,20,21,22,23,25, 27,29,30,31,32,34,37,38,39,41
Probe	3,5,6,23,24,32,33,38
DOS	1,3,6,8,19,23,28,32,33,35,36,38,39,41
U2R	5,6,15,16,18,25,32,33,38,39
R2L	3,5,6,24,32,33,34,35,38

Table 2 shows the results of the classification performance and computing time for training and testing of data by the proposed system using all features. Table 3 shows the results of the classification performance and computing time for training and testing data by the proposed system using extracted features. We can see that using important features for classification gives similar accuracies compared to using all features and reduces the training and testing time. Comparing Table 2 with Table 3, we obtain the following results. The performance when using the extracted features does not show any significant differences to when using all features. This means that the proposed on-line feature extraction method has a good performance in extracting features. The proposed method has another merit in memory requirement. The advantage of the proposed feature extraction method is more efficient in terms of memory requirement than a batch KPCA because the proposed feature extraction method does not require the whole N X N kernel matrix where N is the number of the training data. A second one is that the proposed on-line feature extraction method performance is comparable in performance to a batch KPCA.

Table 2. Performance of proposed system using all features

Class	Accuracy	Training Time (Sec)	Testing Time (Sec)
Normal	98.55	5.83	1.45
Probe	98.59	28.0	1.96
DOS	98.10	16.62	1.74
U2R	98.64	2.7	1.34
R2L	98.69	7.8	1.27

Table 3. Performance of proposed system using extracted features

Class	Accuracy	Training Time (Sec)	Testing Time (Sec)
Normal	98.43	5.25	1.42
Probe	98.63	25.52	1.55
DOS	98.14	15.92	1.48
U2R	98.64	2.17	1.32
R2L	98.70	7.2	1.08

5.4 Suitable for Real-time IDS

Table 2 and Table 3 show that using extracted features decreases the training and testing time compared to using all features. Furthermore classification accuracy of the proposed system is similar to using all features. This makes the proposed IDS suitable for real-time IDS.

5.5 Comparison with SVM

Recently SVM has been a powerful methodology for solving problems in nonlinear classification. To evaluate the classification accuracy of the proposed system it is desirable to compare with SVM. Generally a disadvantage of the incremental method is its accuracy compared to the batch method even though it has the advantage of memory efficiency. According to Table 4 and Table 5 we can see that the proposed method has better classification performance compared to batch SVM. Through this result we can show that the proposed real-time IDS has remarkable classification accuracy, although it is worked in an incremental way.

Table 4. Performance comparison of proposed method and SVM using all features

	Normal	Probe	DOS	U2R	R2L
Proposed method	98.76	98.81	98.56	98.92	98.86
SVM	98.55	98.70	98.25	98.87	98.78

Table 5. Performance comparison of proposed method and SVM using extracted features

	Normal	Probe	DOS	U2R	R2L
Proposed method	98.67	98.72	98.56	98.88	98.78
SVM	98.59	98.38	98.22	98.87	98.78

6. Conclusion and Remarks.

This paper was devoted to the exposition of a new technique on real-time IDSs. To develop this system, we made use of empirical kernel mapping with incremental learning via the eigenspace approach. Proposed on-line KPCA has the following advantages. Firstly, the performance of using the extracted features does not show significant differences to that of using all features. This means that the proposed on-line feature extraction method shows good performance in extracting features. Secondly, the proposed method has merit in memory requirement.

The advantage of the proposed feature extraction method is more efficient in terms of memory requirement than a batch KPCA because the proposed feature extraction method does not require the whole N X N kernel matrix where N is the number of the training data. Thirdly, the proposed on-line feature extraction method is comparable in performance to a batch KPCA, although it works incrementally. Our ongoing experiment involves applying the proposed system to more realistic world data to evaluate the real-time detection performance.

References

[1] I.T. Jolliffe, "Principal Component Analysis", New York Springer-Verlag, 1986.

[2] S. Chandrasekaran, B.S. Manjunath, Y.F. Wang, J. Winkeler and H. Zhang, "An eigenspace update algorithm for image analysis," Graphical Models and Image Processing, 59(5), pp.321-332, Sep., 1997.

[3] J. Winkeler, B.S. Manjunath and S. Chandrasekaran, "Subset selection for active object recognition," In CVPR, volume 2, pp.511-516, IEEE Computer Society Press, Jun., 1999.

[4] P. Hall, D. Marshall, and R. Martin, "Incremental eigenanalysis for classification," In British Machine Vision Conference, volume 1, pp.286-295, Sep., 1998.

[5] M.E. Tipping and C.M. Bishop, "Mixtures of probabilistic principal component analysers," Neural Computation 11(2), pp.443-482, 1998.

[6] M.A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," AICHE Journal 37(2), pp.233-243, 1991.

[7] K.I. Diamantaras and S.Y. Kung, "Principal Component Neural Networks: Theory and Applications", New York John Wiley & Sons, Inc., 1996.

[8] B. Scholkopf, A. Smola and K.R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," Neural Computation 10(5), pp.1299-1319, 1998.

[9] R. Rosipal and M. Girolami, "An Expectation Maximization approach to nonlinear component analysis," Submitted

[10] B. Scholkopf, S. Mika, C. Burges, P. Knirsch, K.R. Miller, G. Ratsch and A.J. Smola, "Input Space versus Feature Space in Kernel-Based Methods," IEEE Transactions on Neural Networks, vol. 10, pp.1000-1017, Sep., 1999.

[11] A.J. Smola, O.L.Mangasarian, and B. Scholkopf, "Sparse kernel feature analysis," Technical Report 99-03, University of Wisconsin, Data Mining Institute, Madison, 1999.

[12] S.R. Gunn, "Support vector machines for classification and regression," Technical Report ISIS-1-98, Department of Electronics and Computer Science, University of Southampton, 1998.

[13] S. Mika, "Kernel algorithms for nonlinear signal processing in feature spaces," Master's thesis, Technical University of Berlin, November. 1998.

[14] H. Murakami and V. Kumar, "Efficient Calculation of Primary Images from a Set of Images," IEEE Trans. On Pattern Analysis and Machine Intelligence, 4 (5), pp.511-515, 1982.

[15] V. N. Vapnik, "Statistical learning theory", John Wiley & Sons, New York, 1998.

[16] K. Tsuda, "Support vector classifier based on asymmetric kernel function," Proc. ESANN, 1999.

[17] J.A.K. Suykens and Vandewalle, "Least squares support vector machine classifiers," Neural Processing Letters, vol.9, pp.293-300, 1999.

[18] Accessible at http://www.esat.kuleuven.ac.be/sista/lssvmlab/tutorial.

Byung-Joo Kim

He received the B.S. degrees in Department of Computer Science from Pusan National University, Busan, Korea and the M.S. degrees in Department of Computer Science from Pusan National University and Ph.D. degrees from Kyungpook National University Daegu, Korea major in computer science. From 2003 to present, he is an Assistant Professor, Department of Information & Communication Engineering at Youngsan University in Korea. His research interests include Machine Learning and Biometric Identification.

Il-Kon Kim

He received the Ph.D. degrees in Departmrnt of Computer Science from Seoul National University, Seoul, Korea. From 1992 to present, he is a Professor, Department of Computer Science at Kyungpook National University Daegu, Korea. From 1997 to 1998 He was a visiting professor at Medical Center in Georgetown University. Now he is a head of Intelligent Clinical Support and Information Sharing Center. His research interests include Intelligent Agent System, Distributed System and Medical Informatics.